REMARKS

Reconsideration of the instant application is respectfully requested. The present submission is responsive to the Office Action of December 7, 2007, in which claims 1-9 and 11 are presently pending. Of those, claims 1, 9 and 11 have been rejected under 35 U.S.C. §112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter regarded as the invention. With regard to the art of record, claims 1-6, 8, 9 and 11 have now been rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent 6,442,752 to Jennings, et al., in view of U.S. Patent 6,295,642 to Blandy. In addition, claim 7 has been rejected under 35 U.S.C. §103(a) as being unpatentable over Jennings in view of Blandy, and further in view of U.S. Patent 6,735,598 to Srivastava, et al. For the following reasons, however, it is respectfully submitted that the application is now in condition for allowance.

Rejections under 35 U.S.C. §112, second paragraph:

Turning first to the rejections under 35 U.S.C. §112, second paragraph, the Examiner has taken the position that the claim language "compiling and linking said source file iteratively to create a single executable file" is indefinite. In particular, the Examiner states in paragraph 6 of the office action that "the 'compiling and linking said source file iteratively' feature is only mentioned in the Abstract, (and it is) not clear to the Examiner what (the reason is) to do the compiling and linking 'iteratively' in order to create a single executable file…"

The answer to the query posed by the Examiner, "[i]s that meant to do compiling and linking and create a single executable for each of the variations, characteristics, and parameters for each attribute?" is affirmative and, moreover, Applicants respectfully submit that such an interpretation would be apparent to one skilled in the art.

The iterations take place over the compilations, and thus a differentiating naming convention need be applied there. However, no formal convention need be established by present invention, the only requirement is that they are expected to be different as this is typically a requirement normally imposed by linkage editors. There need not be any fixed naming convention as the attributes will provide the differentiation required. There is, over the iterations of compilation, a single executable produced, which is a key concept of the present invention.

Notwithstanding, claim 1 has been amended as set forth above to more clearly point out a purpose of iteratively compiling and linking a source file to create a single executable file. In particular, claim 1 now recites in part:

"...compiling said source file iteratively to create corresponding object files based on said at least one of variations, characteristics, and parameters for each said attribute; and

linking a plurality of intermediately resulting object files to create a single executable file..."

In other words, each time the source file is compiled (i.e., an iteration), an object file results. Then, the resulting object files are linked so as to create a single executable file.

Support for this amendment is found at least in specification paragraph [0028], as well as in Figures 2 and 4 of the drawings. For example, in Figure 4, the single DLL executable 54 includes two versions of three functions: abc, abc', def, def', ghi, ghi'.

Accordingly, it is respectfully submitted that the §112, second paragraph rejections have been addressed and overcome.

Rejections under 35 U.S.C. §103:

Turning next to the rejections under 35 U.S.C. §103, the Examiner has taken the position that claims 1-6, 8, 9 and 11 are unpatentable over the combination of the

Jennings and Blandy references. The Applicants respectfully traverse the same for the reason that the combination of the teaching of Jennings and Blandy does not result in each and every element of the claims as presently amended.

First, with regard to the Examiner's statement that the concept of establishing an attribute, with a plurality of definitions via various parameters, is to "overload a method" in a computer program, which is a well known skill to the people in the art at the time of the invention was made, the Applicants concur that "overloading" in and of itself is well known to those skilled in the art. However, the claimed method is quite distinct from conventional overloading. In particular, overloading provides a means whereby a singly named function can be made to correspond to several different implementations, hallmarked by the use of distinctly different parameters. In contrast, the goal of the presently claimed method is to provide versioning of functions with identical signatures. A good example of this would be to produce a single DLL, which provided functions in support of subsequent releases of a given product, where it was critical to retain the older versions for earlier releases of the product. (It should also be noted that the present invention embodiments do not preclude, and in fact allow for, different parameters. However, they are not the means by which the functions or methods are differentiated (versioned)).

Like function overloading, a name mangling technique can be applied in order to allow existing linkage editors, unchanged, to support this method. By the same token, modifications can be made to a linkage editor in support of this method, which allows it (for this invention) to correctly discern and report version mismatches (or to allow some degree of "best-match" capability).

With regard to the Jennings reference, there are numerous implementations taught therein which provide a means to "glue" references from a program to distinct functions based on attributes, by providing additional information as an adjunct to said functions, and then by means of post-processing and/or with run-time support to select the appropriate functions or methods. However, the present invention is different in that it

instead uses information, which can be processed directly by existing linkage editors without further changes. Jennings teaching the ability to select among dynamic link libraries; whereas, whereas the presently claimed invention provides, in a single multi-attribute DLL, all versions of any given function.

The Examiner has acknowledged that Jennings does not teach compiling and linking iteratively, and thus relies on the Blandy reference in making the present §103 rejections. More specifically, the Examiner points out the Abstract of Blandy, which states:

"An iterative process is employed whereby bytecodes are compiled up to the next conditional flow bytecode or return, the compiled code is executed and any attempt to enter uncompiled paths of the method is monitored. When the executing thread attempts to execute an uncompiled path control is returned to the compiler and the process is repeated starting with the first bytecode of that path."

However, this use of the term "iteration" has absolutely nothing to do with the compiling iterations described in the present claims. Rather, Blandy simply describes a method of just-in-time compilation, which uses iteration to compile only the subset of code currently required. In other words, Blandy teaches an iterative compilation method which is built-in to the run-time execution. This is completely different than the claimed invention, which is directed toward a method by which a user iteratively compiles during the process of creating the DLL packages.

Accordingly, it is respectfully submitted that since the combination of Jennings and Blandy does not result in each element of the claims as presently amended, the §103 rejections have been addressed and overcome.

Notwithstanding the above, with respect to claim 4, the Applicants also point out that, unlike Jennings, the differentiations provided for are versioning of a single specific operating system. The claimed "machine architecture" is for different levels of support

provided by advances in the generations of a given machine. Jennings is instead providing a "glue code" method ("jacket routines" which allows for "parameter mapping"). The claimed method does not imply any of the overheads inherent in Jennings' teachings.

Finally, with respect to claim 5, the Applicants further submit that it is incorrect to correlate Jennings' discussion of parameters to the claimed "wherein said attribute is user specified." Jennings is teaching a method by which the user is effectively automating the writing of the software to accommodate differing target machine platforms. In contrast, the claimed method is inherently a packaging methodology, where the user could easily and likely be a product builder. Such a user is not necessarily involved with, or need understand, the intricacies of actually writing the software. To be clear, the present invention does not prohibit that differing parameters can be used with different variations, however they are not themselves the differentiators. This is a key point because the linkage editor does not inspect the parameters and thus would not be able to make selections based upon those differences.

Claims 9 and 11 have been cancelled without prejudice, however the Applicants respectfully reserve the right to pursue the subject matter of the same in one or more continuing applications.

For the above stated reasons, it is respectfully submitted that the present application is now in condition for allowance. No new matter has been entered and no additional fees are believed to be required. However, if any fees are due with respect to this Amendment, please charge them to Deposit Account No. 09-0463 maintained by Applicants' attorneys.

Respectfully submitted,
ANUJA DEEDWANIYA, ET AL.

CANTOR COLBURN LLP
Applicants' Attorneys

By /Sean F. Sullivan/
Sean F. Sullivan
Registration No. 38,328
Customer No. 46429

Date: March 7, 2008
Address: 20 Church Street, Hartford, CT 06103
Telephone: (860) 286-2929